Termination Resilience Analysis

MPRI 2-6: Abstract Interpretation, Application to Verification and Static Analysis

Caterina Urban

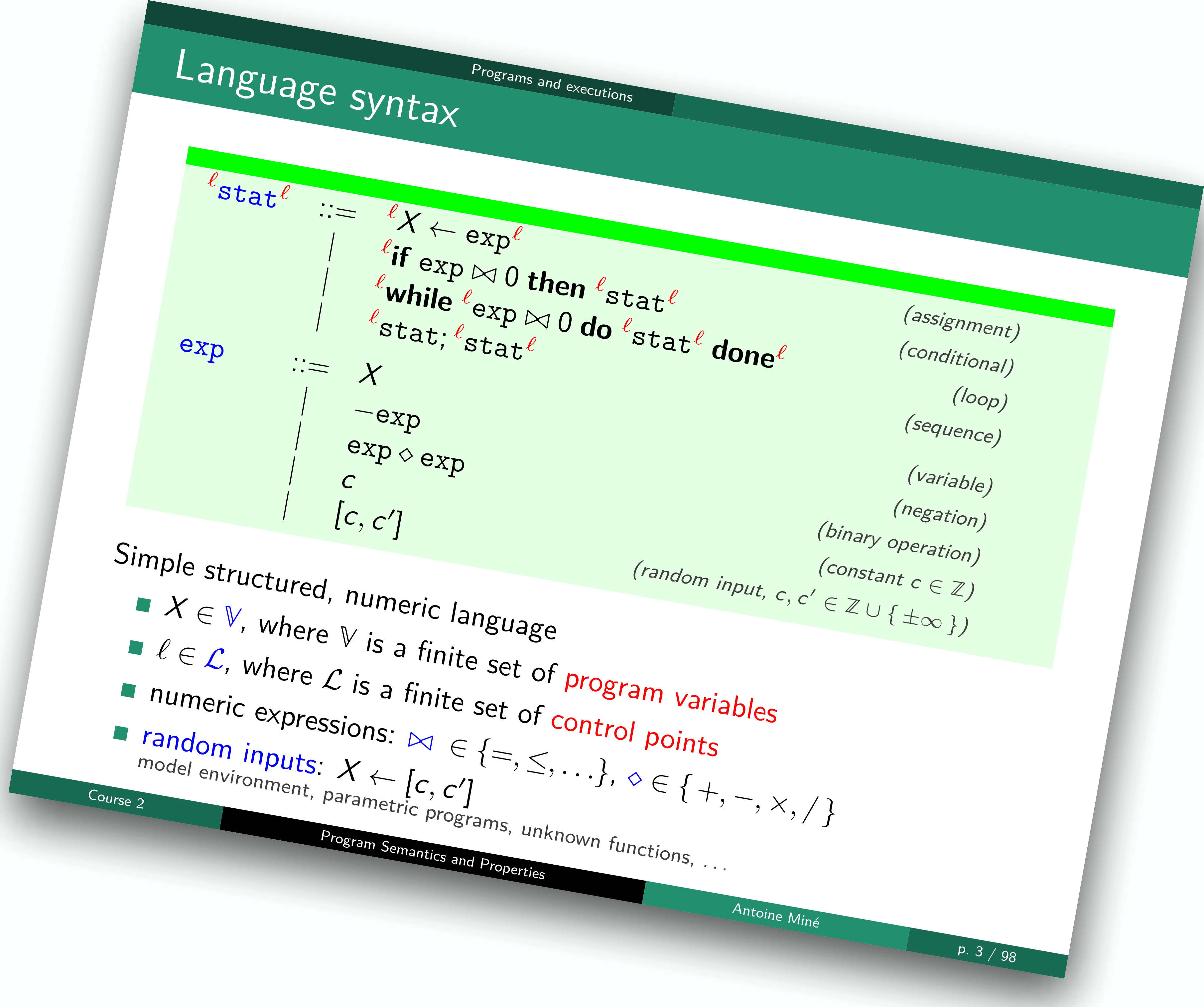
November 18th, 2024

Year 2024-2025

Language Syntax

Example

$$
\begin{aligned}
\text{stat} \; ::= \; & {}^{\ell}X \leftarrow \textbf{input} \\
& {}^{\ell}X \leftarrow \text{exp} \\
& \textbf{if } {}^{\ell}exp \bowtie 0 \textbf{ then } \text{stat} \\
& \textbf{while } {}^{\ell}exp \bowtie 0 \textbf{ do } \text{stat } \textbf{done} \\
& \text{stat} \; ; \; \text{stat}
\end{aligned}
$$

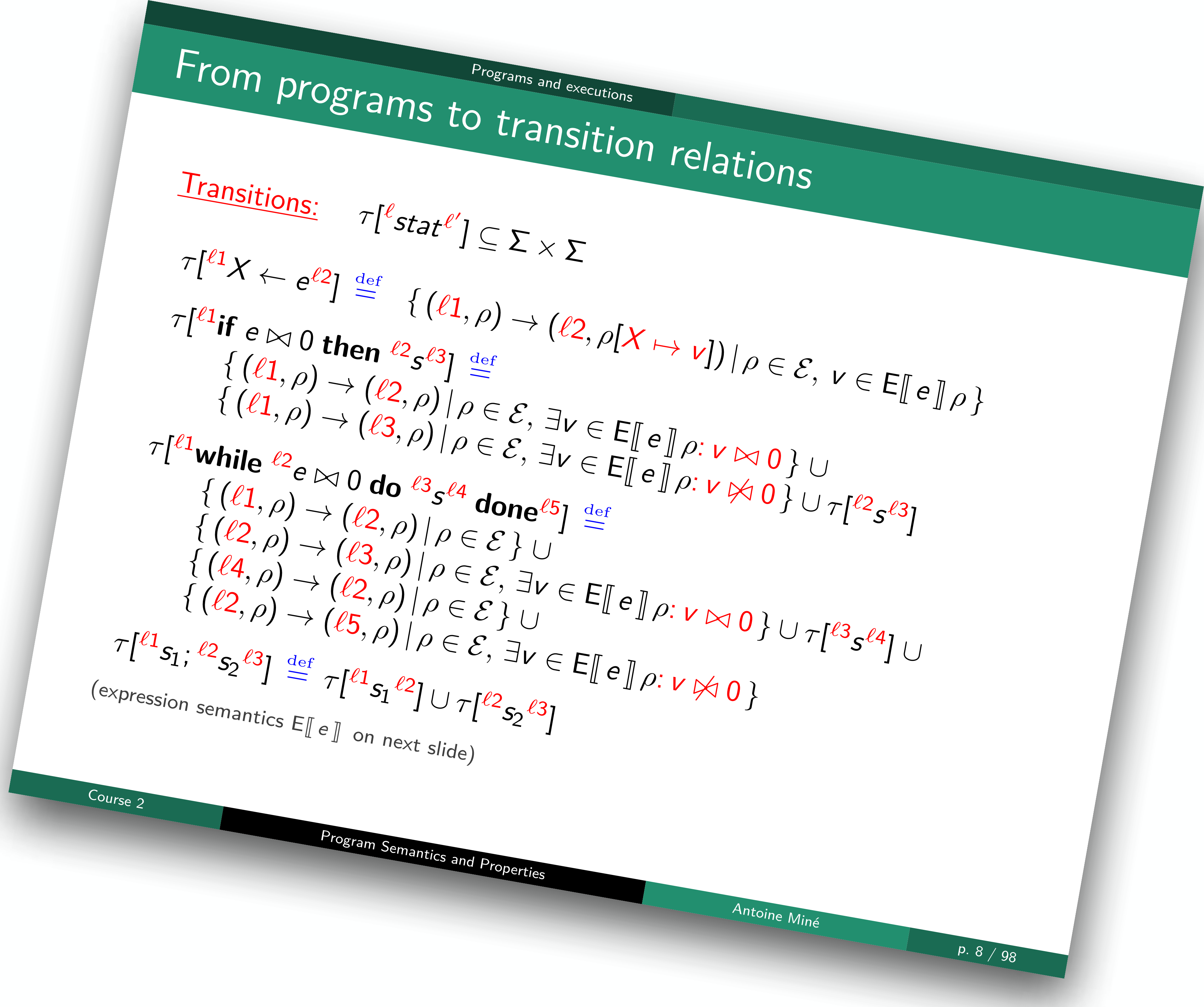


^{1}x ← [-∞, +∞]
^{2}y ← **input**
while ^{3}(x > 0) **do**
    ^{4}x ← x - y
od^{5}

Transition Systems

Example

^{1}x ← [-∞, +∞]
^{2}y ← **input**
while ^{3}(x > 0) **do**
    ^{4}x ← x - y
od^{5}



$$\Sigma \stackrel{\text{def}}{=} \{1,2,3,4,5\} \times \mathscr{E}$$

$$\tau \stackrel{\text{def}}{=} \{(\mathbf{1}, \rho) \to (\mathbf{2}, \rho[x \mapsto v]) \mid \rho \in \mathscr{E}, v \in \mathbb{Z}\}$$
$$\cup\ \{(\mathbf{2}, \rho) \to (\mathbf{3}, \rho[y \mapsto v]) \mid\ \mid \rho \in \mathscr{E}, v \in \mathbb{Z}\}$$
$$\cup\ \{(\mathbf{3}, \rho) \to (\mathbf{4}, \rho) \mid\ \mid \rho \in \mathscr{E}, \exists v \in E[\![x]\!]\rho : v > 0\}$$
$$\cup\ \{(\mathbf{4}, \rho) \to (\mathbf{3}, \rho[x \mapsto v]) \mid \rho \in \mathscr{E}, v \in E[\![x - y]\!]\rho\}$$
$$\cup\ \{(\mathbf{3}, \rho) \to (\mathbf{5}, \rho) \mid\ \mid \rho \in \mathscr{E}, \exists v \in E[\![x]\!]\rho : v \leq 0\}$$

Termination Resilience

Potential Termination

Definition

A program with trace semantics $\mathscr{M} \in \mathscr{P}(\Sigma^\infty)$ **may terminate** if and only if $\mathscr{M} \cap \Sigma^* \neq \emptyset$

Definite Termination

Definition

A program with trace semantics $\mathscr{M} \in \mathscr{P}(\Sigma^\infty)$ **may terminate** if and only if $\mathscr{M} \cap \Sigma^* \neq \emptyset$

$$\text{PotentialTermination} \stackrel{\text{def}}{=} \{T \in \mathscr{P}(\Sigma^\infty) \mid \exists t \in T\colon\ t \in \Sigma^*\}$$

$$\text{DefiniteTermination} \stackrel{\text{def}}{=} \{T \in \mathscr{P}(\Sigma^\infty) \mid \forall t \in T\colon\ t \in \Sigma^*\}$$

input values read in a traces t

$$\text{TerminationResilience} \stackrel{\text{def}}{=} \{T \in \mathscr{P}(\Sigma^\infty) \mid \forall i \in T^i \exists t \in T\colon t^i = i \wedge t \in \Sigma^*\}$$

input values read in a set of traces T

Termination Resilience

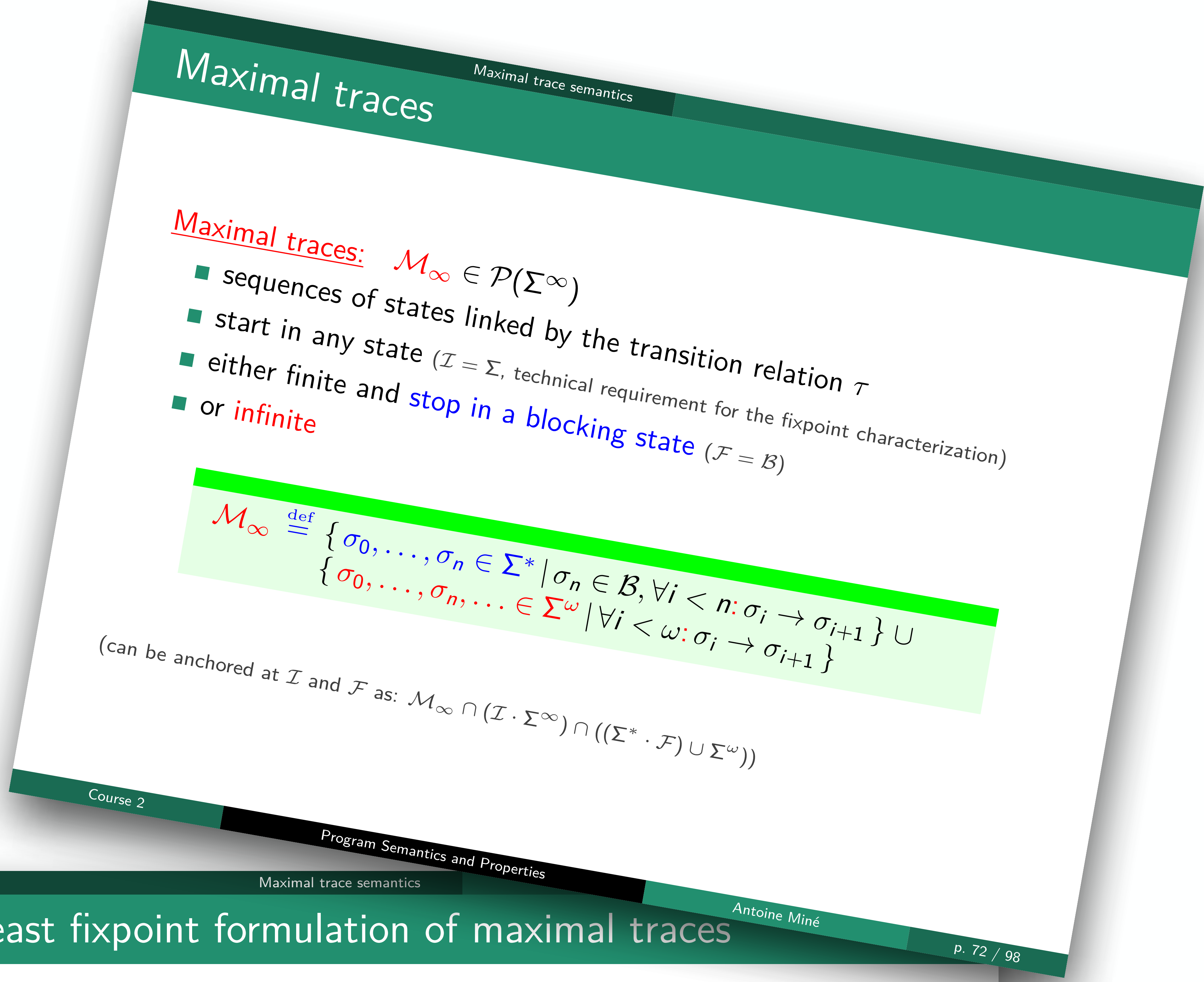
Example

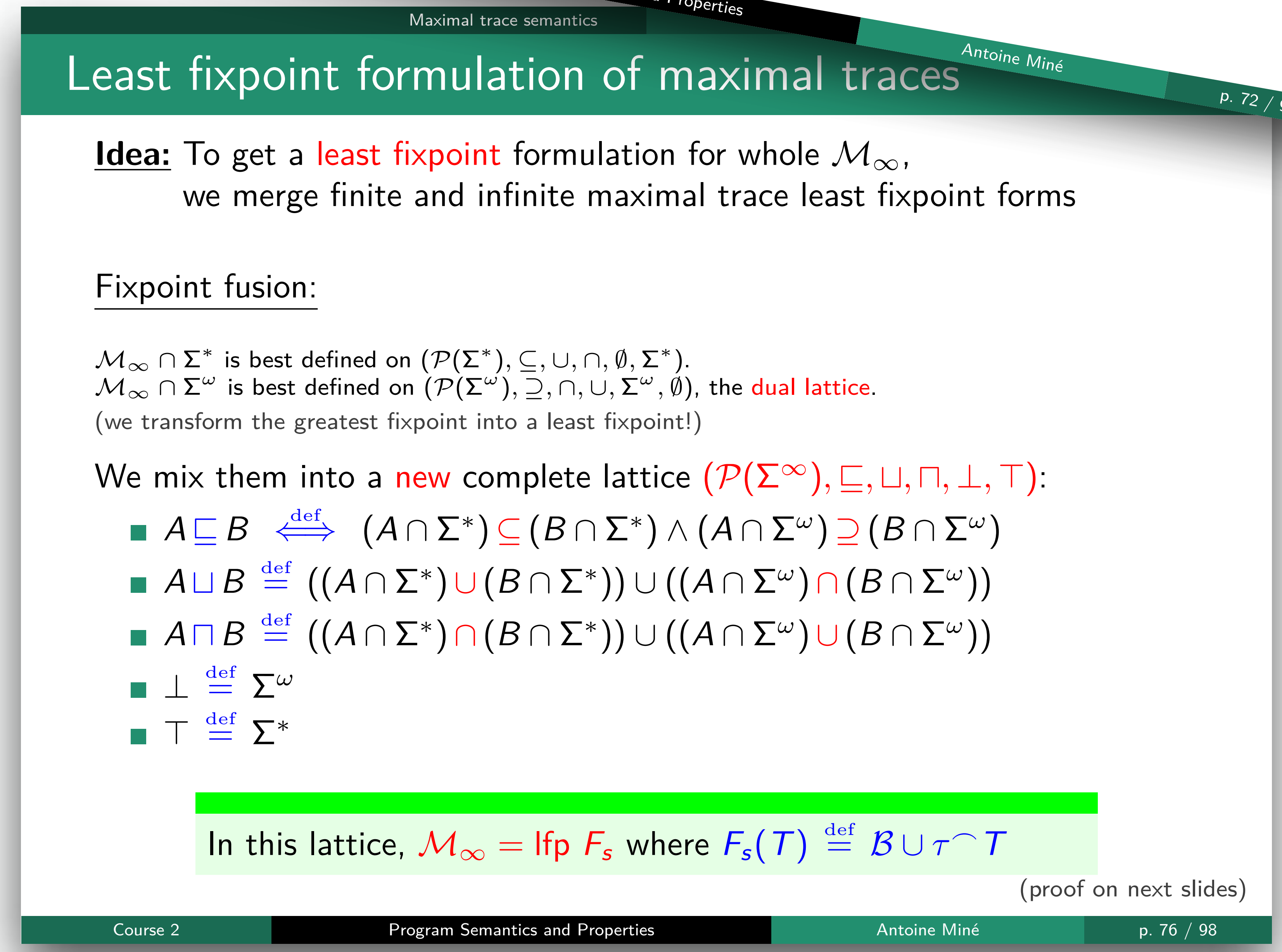
^{1}x ← [-∞, +∞]
^{2}y ← input
while ^{3}(x > 0) **do**
^{4}x ← x - y
od^5

$\mathscr{M}_\infty \in$ TerminationResilience

$y > 0$: the program always terminates

$y \leq 0$: the program terminates when $x \leq 0$





Termination Resilience Semantics

practical tools
targeting specific programs

algorithmic approaches
to decide program properties

mathematical models
of the program behavior

Termination Resilience Semantics

Potential Termination Semantics

$$\mathcal{R}_m = \text{lfp}^{\preceq} F_m$$

$$F_m(f)\sigma \stackrel{\text{def}}{=} \begin{cases} 0 & \sigma \in \mathcal{B} \\ v & \sigma \in \text{pre}_\tau(\text{dom}(f)) \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$v \stackrel{\text{def}}{=} \inf\{f(\sigma') + 1 \mid (\sigma, \sigma') \in \tau\}$$

Definite Termination Semantics

$$\mathcal{R}_M = \text{lfp}^{\preceq} F_M$$

$$F_M(f)\sigma \stackrel{\text{def}}{=} \begin{cases} 0 & \sigma \in \mathcal{B} \\ \overline{v} & \sigma \in \widetilde{\text{pre}}_\tau(\text{dom}(f)) \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\overline{v} \stackrel{\text{def}}{=} \sup\{f(\sigma') + 1 \mid (\sigma, \sigma') \in \tau\}$$

$$\mathcal{R}_{TR} = \text{lfp}^{\preceq} F_{TR}$$

$$F_{TR}(f)\sigma \stackrel{\text{def}}{=} \begin{cases} 0 & \sigma \in \mathcal{B} \\ \overline{v} & \sigma \in \widetilde{\text{pre}}_{\tau^i}(\text{dom}(f)) \\ v & \sigma \in \text{pre}_{\tau^r}(\text{dom}(f)) \\ \text{undefined} & \text{otherwise} \end{cases}$$

input state transitions

regular state transitions

Denotational Termination Resilience Semantics

For each program instruction stat, we define a transformer $\mathscr{R}_{TR}[\![\text{stat}]\!] : (\mathscr{E} \rightharpoonup \mathbb{O}) \to (\mathscr{E} \rightharpoonup \mathbb{O})$:

- $\mathscr{R}_{TR}[\![{}^{\ell}X \leftarrow \textbf{input}]\!]$
- $\mathscr{R}_{TR}[\![{}^{\ell}X \leftarrow e]\!]$
- $\mathscr{R}_{TR}[\![\textbf{if } {}^{\ell}\, e \bowtie 0 \textbf{ then } s]\!]$
- $\mathscr{R}_{TR}[\![\textbf{while } {}^{\ell}\, e \bowtie 0 \textbf{ do } s \textbf{ done}]\!]$
- $\mathscr{R}_{TR}[\![s_1; s_2]\!]$

Denotational Termination Resilience Semantics

$\mathcal{R}_{TR}[\![{}^{\ell}X \leftarrow \textbf{input}]\!]$

$$\mathcal{R}_{TR}[\![{}^{\ell}X \leftarrow \textbf{input}]\!]f \stackrel{\text{def}}{=} \lambda\rho\,.\begin{cases} \sup\{f(\rho[X \mapsto v]) + 1 \mid v \in E[\![e]\!]\rho\} & \forall v \in \mathbb{Z}: \rho[X \mapsto v] \in \text{dom}(f) \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\mathcal{R}_{TR} = \text{lfp}^{\preceq} F_{TR}$$

$$F_{TR}(f)\sigma \stackrel{\text{def}}{=} \begin{cases} 0 & \sigma \in \mathcal{B} \\ \overline{v} & \sigma \in \widetilde{\text{pre}}_{\tau^i}(\text{dom}(f)) \\ v & \sigma \in \text{pre}_{\tau^r}(\text{dom}(f)) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Denotational Termination Resilience Semantics

$\mathcal{R}_{TR}[\![{}^{\ell}X \leftarrow e]\!]$

the value of e (and thus of X) depends from the input values read by the program

$$\mathcal{R}_{TR}[\![{}^{\ell}X \leftarrow e]\!]f \stackrel{\text{def}}{=} \lambda\rho\,.\begin{cases} \text{sup}\{f(\rho[X \mapsto v]) + 1 \mid v \in E[\![e]\!]\rho\} & \mathcal{T}[\![e]\!]\mathcal{T}(\ell) \wedge \text{all} \\ \text{inf}\{f(\rho[X \mapsto v]) + 1 \mid v \in E[\![e]\!]\rho\} & \neg\mathcal{T}[\![e]\!]\mathcal{T}(\ell) \wedge \text{any} \\ \text{undefined} & \text{otherwise} \end{cases}$$

$$\text{all} \stackrel{\text{def}}{=} \exists \bar{v} \in E[\![e]\!]\rho \wedge \forall v \in E[\![e]\!]\rho \colon \rho[X \mapsto v] \in \text{dom}(f)$$

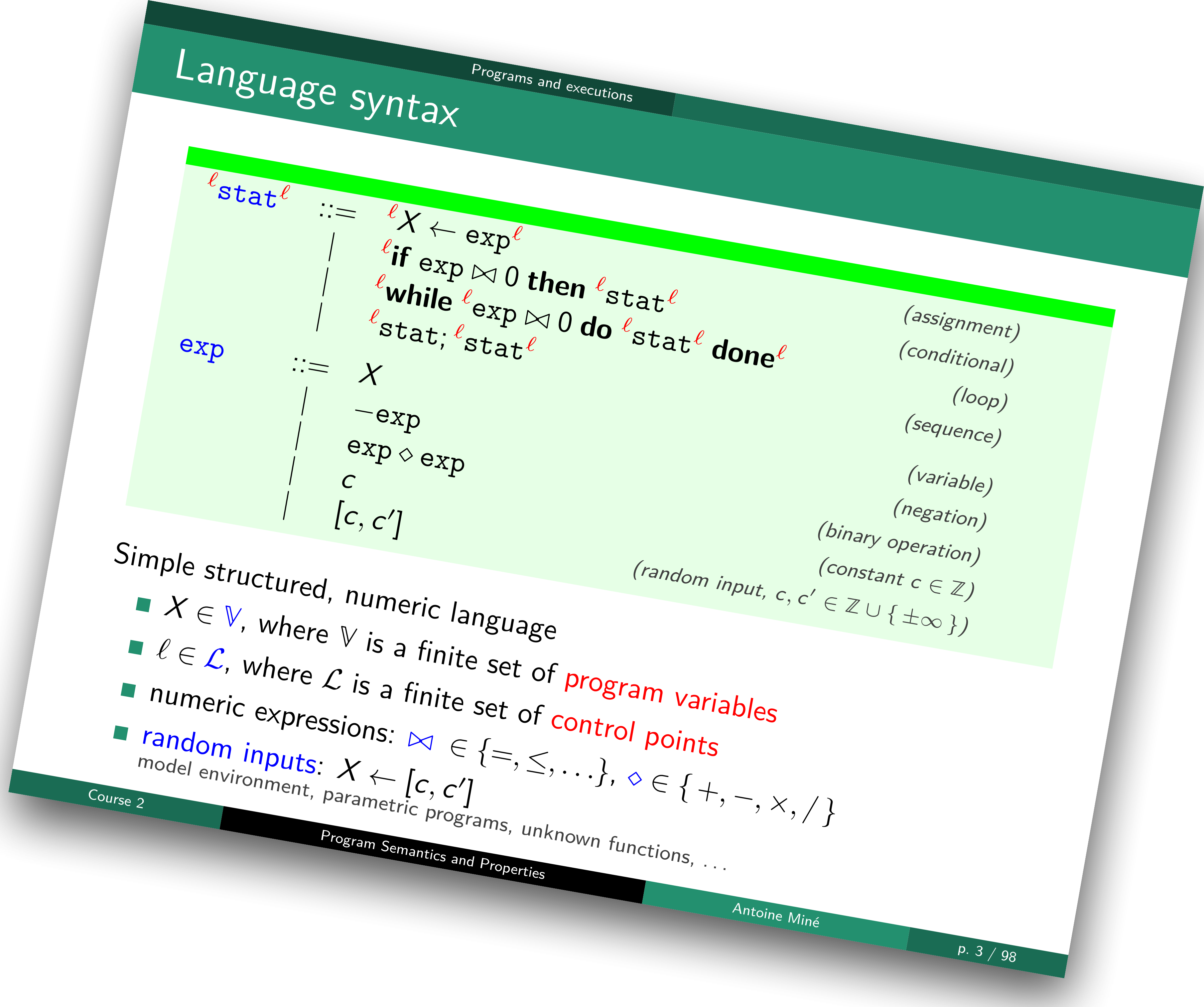
$$\text{any} \stackrel{\text{def}}{=} \exists v \in E[\![e]\!]\rho \colon \rho[X \mapsto v] \in \text{dom}(f)$$

$$\mathcal{R}_{TR} = \text{lfp}^{\preceq} F_{TR}$$

$$F_{TR}(f)\sigma \stackrel{\text{def}}{=} \begin{cases} 0 & \sigma \in \mathcal{B} \\ \bar{v} & \sigma \in \widetilde{\text{pre}}_{\tau^i}(\text{dom}(f)) \\ v & \sigma \in \text{pre}_{\tau^r}(\text{dom}(f)) \\ \text{undefined} & \text{otherwise} \end{cases}$$

Simple Taint Semantics

$\mathcal{T}[\![\text{stat}]\!] : \mathcal{P}(\mathbb{X}) \to \mathcal{P}(\mathbb{X})$



$\mathcal{T}[\![{}^{\ell}X \leftarrow \textbf{input}]\!]T \stackrel{\text{def}}{=} T \cup \{X\}$

$\mathcal{T}[\![{}^{\ell}X \leftarrow e]\!]T \stackrel{\text{def}}{=} \begin{cases} T \cup \{X\} & \mathcal{T}[\![e]\!]T \\ T \setminus \{X\} & \text{otherwise} \end{cases}$

assigned variables in s

$\mathcal{T}[\![\textbf{if}\ {}^{\ell}e \bowtie 0\ \textbf{then}\ s]\!]T \stackrel{\text{def}}{=} \begin{cases} \mathcal{T}[\![s]\!]T \cup \text{LHS}(s) \cup T & \mathcal{T}[\![e]\!]T \\ \mathcal{T}[\![s]\!]T \cup T & \text{otherwise} \end{cases}$

$\mathcal{T}[\![\textbf{while}\ {}^{\ell}e \bowtie 0\ \textbf{do}\ s\ \textbf{done}]\!]T \stackrel{\text{def}}{=} \text{lfp}^{\subseteq}_{T} \left(\lambda X.\begin{cases} \mathcal{T}[\![s]\!]X \cup \text{LHS}(s) \cup X & \mathcal{T}[\![e]\!]X \\ \mathcal{T}[\![s]\!]X \cup X & \text{otherwise} \end{cases} \right)$

$\mathcal{T}[\![s_1; s_2]\!]T \stackrel{\text{def}}{=} \mathcal{T}[\![s_2]\!](\mathcal{T}[\![s_1]\!]T)$

$\mathcal{T}[\![e]\!]T \stackrel{\text{iff}}{\Leftrightarrow} \exists \rho \in \mathcal{E} : \exists v \in E[\![e]\!]\rho : \forall \rho' \in \mathcal{E} : \rho' \neq_T \rho \Rightarrow v \notin E[\![e]\!]\rho'$

$\rho' \neq_T \rho \equiv \forall x \notin T : \rho'(x) = \rho(x) \wedge \exists x \in T : \rho'(x) \neq \rho(x)$

Denotational Termination Resilience Semantics

$\mathcal{R}_{TR}[\![\mathbf{if}\ {}^{\ell}\ e \bowtie 0\ \mathbf{then}\ s]\!]$

$$\mathcal{R}_{TR}[\![\mathbf{if}\ {}^{\ell}\ e \bowtie 0\ \mathbf{then}\ s]\!]f \stackrel{\text{def}}{=} \lambda\rho\,.\begin{cases} ① & \exists v_1, v_2 \in E[\![e]\!]\rho : v_1 \bowtie 0 \wedge v_2 \not\bowtie 0 \\ ② & \exists v_1, v_2 \in E[\![e]\!]\rho : v_1 \bowtie 0 \wedge v_2 \not\bowtie 0 \\ \mathcal{R}_{TR}[\![s]\!]f(\rho) + 1 & \rho \in \text{dom}(\mathcal{R}_{TR}[\![s]\!]f) \wedge \forall v \in E[\![e]\!]\rho : v \bowtie 0 \\ f(\rho) + 1 & \rho \in \text{dom}(f) \wedge \forall v \in E[\![e]\!]\rho : v \not\bowtie 0 \\ \text{undefined} & \text{otherwise} \end{cases}$$

① $\sup\{\mathcal{R}_{TR}[\![s]\!]f(\rho) + 1,\ f(\rho) + 1\}$ $\quad \mathcal{T}[\![e]\!]\mathcal{T}(\ell) \wedge \rho \in \text{dom}(\mathcal{R}_{TR}[\![s]\!]f) \cap \text{dom}(f)$

② $\inf\{\mathcal{R}_{TR}[\![s]\!]f(\rho) + 1,\ f(\rho) + 1\}$ $\quad \neg\mathcal{T}[\![e]\!]\mathcal{T}(\ell) \wedge \rho \in \text{dom}(\mathcal{R}_{TR}[\![s]\!]f) \cap \text{dom}(f)$

$\mathcal{R}_{TR}[\![s]\!]f(\rho) + 1$ $\quad \neg\mathcal{T}[\![e]\!]\mathcal{T}(\ell) \wedge \rho \in \text{dom}(\mathcal{R}_{TR}[\![s]\!]f) \setminus \text{dom}(f)$

$f(\rho) + 1$ $\quad \neg\mathcal{T}[\![e]\!]\mathcal{T}(\ell) \wedge \rho \in \text{dom}(f) \setminus \text{dom}(\mathcal{R}_{TR}[\![s]\!]f)$

Denotational Termination Resilience Semantics

$\mathcal{R}_M[\![\textbf{while}\ {}^{\ell}\ e \bowtie 0\ \textbf{do}\ s\ \textbf{done}]\!]$

$$\mathcal{R}_{TR}[\![\textbf{while}\ {}^{\ell}\ e \bowtie 0\ \textbf{do}\ s\ \textbf{done}]\!]f \stackrel{\text{def}}{=} \text{lfp}^{\dot{\preceq}}_{\dot{\emptyset}} \left(\lambda x\, \rho\, .\begin{cases} ① & \exists v_1, v_2 \in E[\![e]\!]\rho : v_1 \bowtie 0 \land v_2 \not\bowtie 0 \\ ② & \exists v_1, v_2 \in E[\![e]\!]\rho : v_1 \bowtie 0 \land v_2 \not\bowtie 0 \\ \mathcal{R}_{TR}[\![s]\!]x(\rho) + 1 & \rho \in \text{dom}(\mathcal{R}_{TR}[\![s]\!]x) \land \forall v \in E[\![e]\!]\rho : v \bowtie 0 \\ f(\rho) + 1 & \rho \in \text{dom}(f) \land \forall v \in E[\![e]\!]\rho : v \not\bowtie 0 \\ \text{undefined} & \text{otherwise} \end{cases} \right)$$

① $\quad \text{sup}\{\mathcal{R}_{TR}[\![s]\!]x(\rho) + 1,\ f(\rho) + 1\} \quad \mathcal{T}[\![e]\!]\mathcal{T}(\ell) \land \rho \in \text{dom}(\mathcal{R}_{TR}[\![s]\!]x) \cap \text{dom}(f)$

② $\quad \text{inf}\{\mathcal{R}_{TR}[\![s]\!]x(\rho) + 1,\ f(\rho) + 1\} \quad \neg\mathcal{T}[\![e]\!]\mathcal{T}(\ell) \land \rho \in \text{dom}(\mathcal{R}_{TR}[\![s]\!]x) \cap \text{dom}(f)$

$\quad \mathcal{R}_{TR}[\![s]\!]x(\rho) + 1 \quad \neg\mathcal{T}[\![e]\!]\mathcal{T}(\ell) \land \rho \in \text{dom}(\mathcal{R}_{TR}[\![s]\!]x) \setminus \text{dom}(f)$

$\quad f(\rho) + 1 \quad \neg\mathcal{T}[\![e]\!]\mathcal{T}(\ell) \land \rho \in \text{dom}(f) \setminus \text{dom}(\mathcal{R}_{TR}[\![s]\!]x)$

Denotational Termination Resilience Semantics

$\mathcal{R}_M[\![s_1; s_2]\!]$

$$\mathcal{R}_{TR}[\![s_1; s_2]\!]f \stackrel{\text{def}}{=} \mathcal{R}_{TR}[\![s_1]\!](\mathcal{R}_{TR}[\![s_2]\!]f)$$

Denotational Termination Resilience Semantics

Definition

The **denotational termination resilience semantics** $\mathscr{R}_{TR}[\![\text{stat}^{\ell}]\!] : \mathscr{E} \rightharpoonup \mathbb{O}$ of a program stat^{ℓ} is:

$$\mathscr{R}_{TR}[\![\text{stat}^{\ell}]\!] \stackrel{\text{def}}{=} \mathscr{R}_{TR}[\![\text{stat}]\!](\lambda\rho.0)$$

where $\mathscr{R}_{TR}[\![\text{stat}]\!] : (\mathscr{E} \rightharpoonup \mathbb{O}) \to (\mathscr{E} \rightharpoonup \mathbb{O})$ is the denotational termination resilience semantics of each program instruction stat

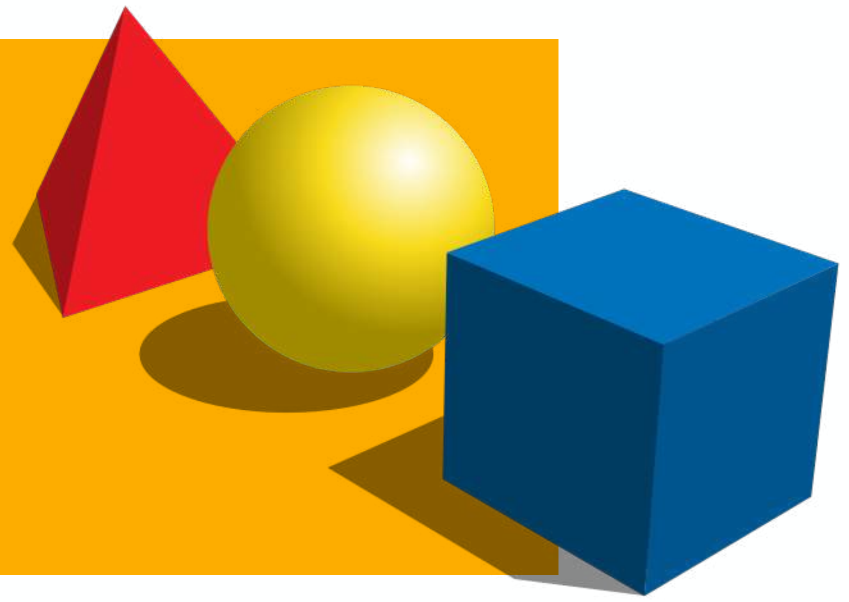
Theorem (Soundness and Completeness)

A program stat^{ℓ} **satisfies termination resilience** for traces starting from a set of initial states $\mathscr{I}$, i.e., $\mathscr{M}_{\infty}[\![\text{stat}^{\ell}]\!](\mathscr{I}) \in \text{TerminationResilience}$, if and only if $\mathscr{I} \subseteq \text{dom}(\mathscr{R}_{TR}[\![\text{stat}^{\ell}]\!])$

Piecewise-Defined Ranking Functions

practical tools
targeting specific programs

algorithmic approaches
to decide program properties



mathematical models
of the program behavior

Concretization-Based Piecewise Abstraction

$\mathcal{R}^{\#}_{TR}[\![\mathsf{stat}^{\ell}]\!] \in \mathcal{A}$

$$\langle \mathcal{E} \rightharpoonup \mathbb{O}, \preccurlyeq \rangle \xleftarrow{\gamma_A} \langle \mathcal{A}, \preccurlyeq_A \rangle$$

$\mathcal{R}_{TR}[\![\mathsf{stat}^{\ell}]\!] \colon \mathcal{E} \rightharpoonup \mathbb{O}$

$f_1 \preccurlyeq f_2 \overset{\text{def}}{=} \text{dom}(f_1) \supseteq \text{dom}(f_2) \wedge \forall x \in \text{dom}(f_1) \colon f_1(x) \leq f_2(x)$

approximation order

By *pointwise lifiting* we obtain an abstraction $\mathcal{R}^{\#}_{TR}$ of $\mathcal{R}_{TR}$:

$$\langle \mathcal{L} \to (\mathcal{E} \rightharpoonup \mathbb{O}), \dot{\preccurlyeq} \rangle \xleftarrow{\dot{\gamma}_A} \langle \mathcal{L} \to \mathcal{A}, \dot{\preccurlyeq}_A \rangle$$

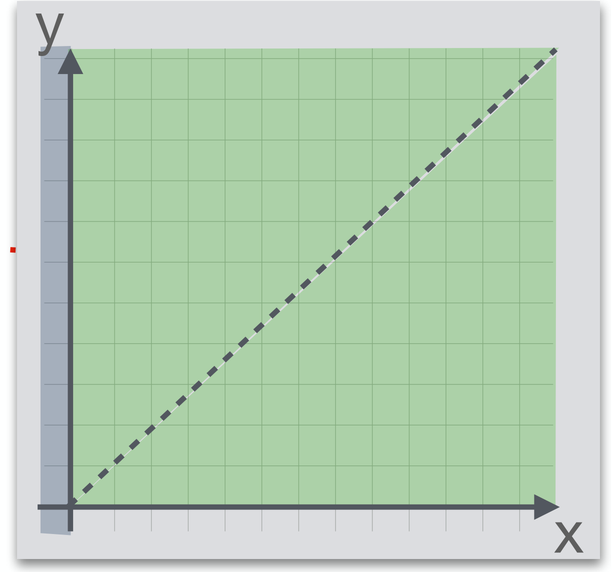
$\mathcal{R}_{TR} \colon \mathcal{L} \to (\mathcal{E} \rightharpoonup \mathbb{O})$

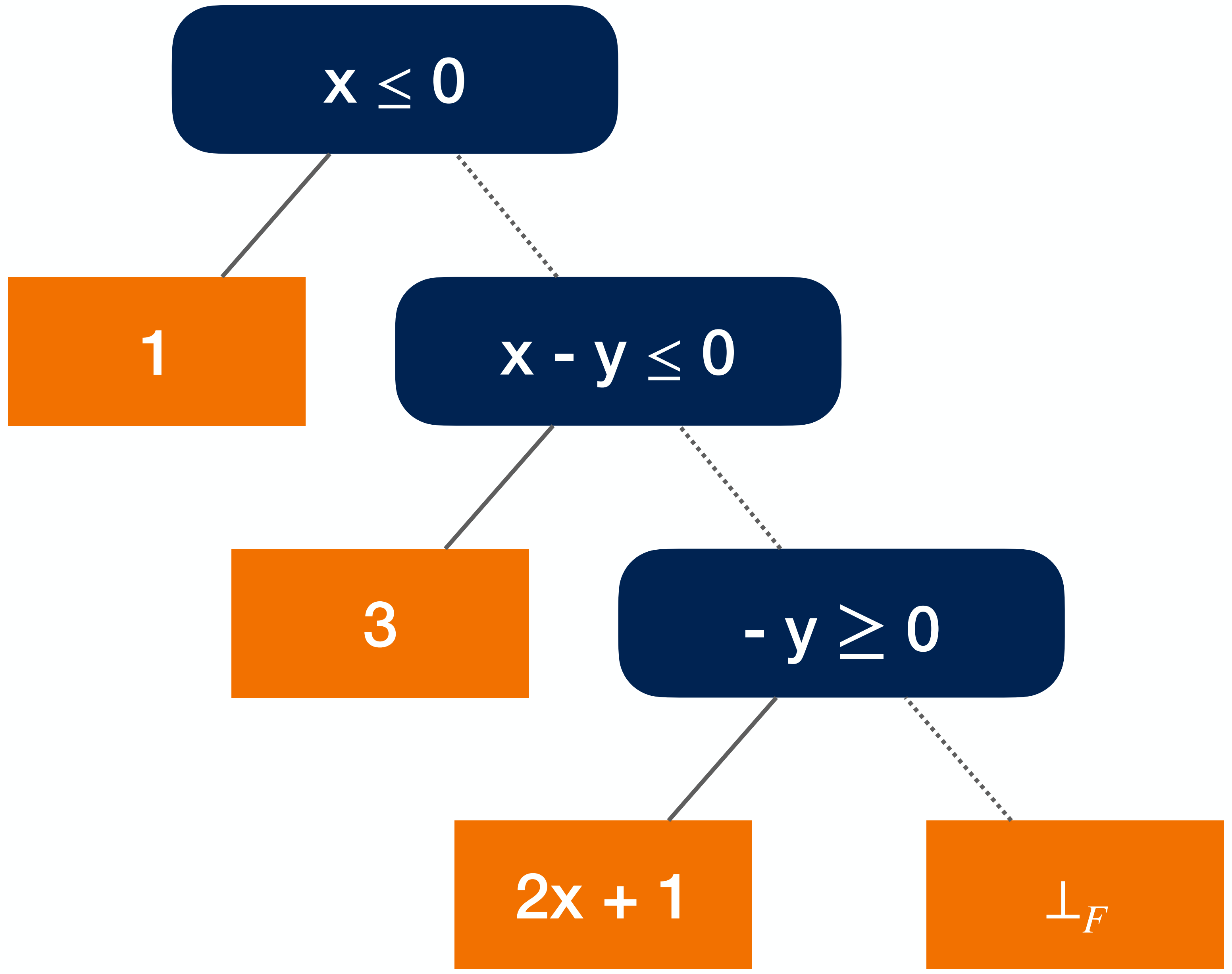
$\mathcal{R}^{\#}_{TR} \colon \mathcal{L} \to \mathcal{A}$

Piecewise-Defined Ranking Functions Abstract Domain

$$\mathcal{A} \stackrel{\text{def}}{=} \{\text{LEAF}: f \mid f \in \mathcal{F}\} \cup \{\text{NODE}\{c\}: t_1; t_2 \mid c \in \mathcal{C} \wedge t_1, t_2 \in \mathcal{A}\}$$

^{1}x ← [-∞, +∞]
^{2}y ← **input**
while ^{3}(x > 0) **do**
  ^{4}x ← x - y
od^5





Piecewise-Defined Ranking Functions Abstract Domain

Functions Auxiliary Abstract Domain

- **computational order** $\sqsubseteq_F [D]$, where $D \in \mathscr{D}$:
  - between defined leaf nodes:

$$f_1 \sqsubseteq_F [D]\ f_2 \overset{\text{def}}{=} \forall \rho \in \gamma_D(D) : f_1(\ldots, \rho(X_i), \ldots) \leq f_2(\ldots, \rho(X_i), \ldots)$$

  - otherwise (i.e., when one or both leaf nodes are undefined):

$$\begin{array}{c} \top_F \\ | \\ f \colon \mathbb{Z}^{|\mathbb{V}|} \rightarrow \mathbb{N} \\ | \\ \bot_F \end{array}$$

Piecewise-Defined Ranking Functions Abstract Domain

Functions Auxiliary Abstract Domain

- **approximation order $\preccurlyeq_F [D]$**, where $D \in \mathscr{D}$:
  - between <u>defined</u> leaf nodes:

$$f_1 \preccurlyeq_F [D]\ f_2 \overset{\text{def}}{=} \forall \rho \in \gamma_D(D) : f_1(\ldots, \rho(X_i), \ldots) \leq f_2(\ldots, \rho(X_i), \ldots)$$

  - otherwise (i.e., when one or both leaf nodes are <u>undefined</u>):

$$\begin{tikzcd} \bot_F & & \top_F \\ & f\colon \mathbb{Z}^{|\mathbb{V}|} \to \mathbb{N} \arrow[ul, dash] \arrow[ur, dash] & \end{tikzcd}$$

Piecewise-Defined Ranking Functions Abstract Domain

Functions Auxiliary Abstract Domain

- **resilience order** $\preccurlyeq_F[D]$, where $D \in \mathscr{D}$:
  - between <u>defined</u> leaf nodes:

$$f_1 \preccurlyeq_F [D]\ f_2 \stackrel{\text{def}}{=} \forall \rho \in \gamma_D(D) : f_1(\ldots, \rho(X_i), \ldots) \geq f_2(\ldots, \rho(X_i), \ldots)$$

  - otherwise (i.e., when one or both leaf nodes are <u>undefined</u>):

$$\begin{tikzcd} & f\colon \mathbb{Z}^{|\mathbb{V}|} \to \mathbb{N} \arrow[dl, dash] \arrow[dr, dash] & \\ \bot_F & & \top_F \end{tikzcd}$$

Piecewise-Defined Ranking Functions Abstract Domain

Abstract Domain Operators

- They manipulate elements in $\mathscr{A}_{\mathsf{NIL}} \stackrel{\text{def}}{=} \{\mathsf{NIL}\} \cup \mathscr{A}$

- The **binary operators** rely on a <u>tree unification</u> algorithm
  - approximation order $\preccurlyeq_A$, computational order $\sqsubseteq_A$, and **resilience order** $\leqslant_A$
  - approximation join $\curlyvee_A$, computational join $\sqcup_A$, and **resilience join** $\underline{\vee}_A$
  - meet $\curlywedge_A$
  - widening $\triangledown_A$

- The **unary operators** rely on a <u>tree pruning</u> algorithm
  - assignment $\overleftarrow{\mathsf{ASSIGN}}_A[\![X \leftarrow e]\!]$
  - test $\mathsf{FILTER}_A[\![e]\!]$

Piecewise-Defined Ranking Functions Abstract Domain

Resilience Order

1. Perform **tree unification**

2. Recursively descend the trees while *accumulating the linear constraints encountered along the paths* into a set of constraints C

3. Compare the leaf nodes using the **resilience order** $\preccurlyeq_F [\alpha_C(C)]$

Piecewise-Defined Ranking Functions Abstract Domain

Resilience Join

1. Perform **tree unification**

2. Recursively descend the trees while *accumulating the linear constraints encountered along the paths* into a set of constraints C

3. $\text{NIL} \curlyvee_A t \stackrel{\text{def}}{=} t$
   $t \curlyvee_A \text{NIL} \stackrel{\text{def}}{=} t$

4. Join the leaf nodes using the **resilience join** $\underline{\vee}_F\,[\alpha_C(C)]$

Piecewise-Defined Ranking Functions Abstract Domain

Resilience Join (continue)

- **resilience join** $\underline{\vee}_F\ [D]$, where $D \in \mathscr{D}$:
  - between <u>defined</u> leaf nodes:

$$f_1 \ \underline{\vee}_F\ [D]\ f_2 \stackrel{\text{def}}{=} \begin{cases} f & f \in \mathscr{F} \setminus \{\ \bot_F\ ,\ \top_F\ \} \\ \top_F & \text{otherwise} \end{cases}$$

where $f \stackrel{\text{def}}{=} \lambda \rho \in \gamma_D(D) :\ \min(f_1(\ldots, \rho(X_i), \ldots), f_2(\ldots, \rho(X_i), \ldots))$

  - otherwise (i.e., when one or both leaf nodes are <u>undefined</u>):

$$\begin{array}{ll} \bot_F \curlyvee_F [D]\ f \stackrel{\text{def}}{=} f & f \in \mathscr{F} \setminus \{\ \top_F\ \} \\ f \curlyvee_F [D]\ \bot_F \stackrel{\text{def}}{=} f & f \in \mathscr{F} \setminus \{\ \top_F\ \} \\ \top_F \curlyvee_F [D]\ f \stackrel{\text{def}}{=} f & f \in \mathscr{F} \setminus \{\ \bot_F\ \} \\ f \curlyvee_F [D]\ \top_F \stackrel{\text{def}}{=} f & f \in \mathscr{F} \setminus \{\ \bot_F\ \} \end{array}$$

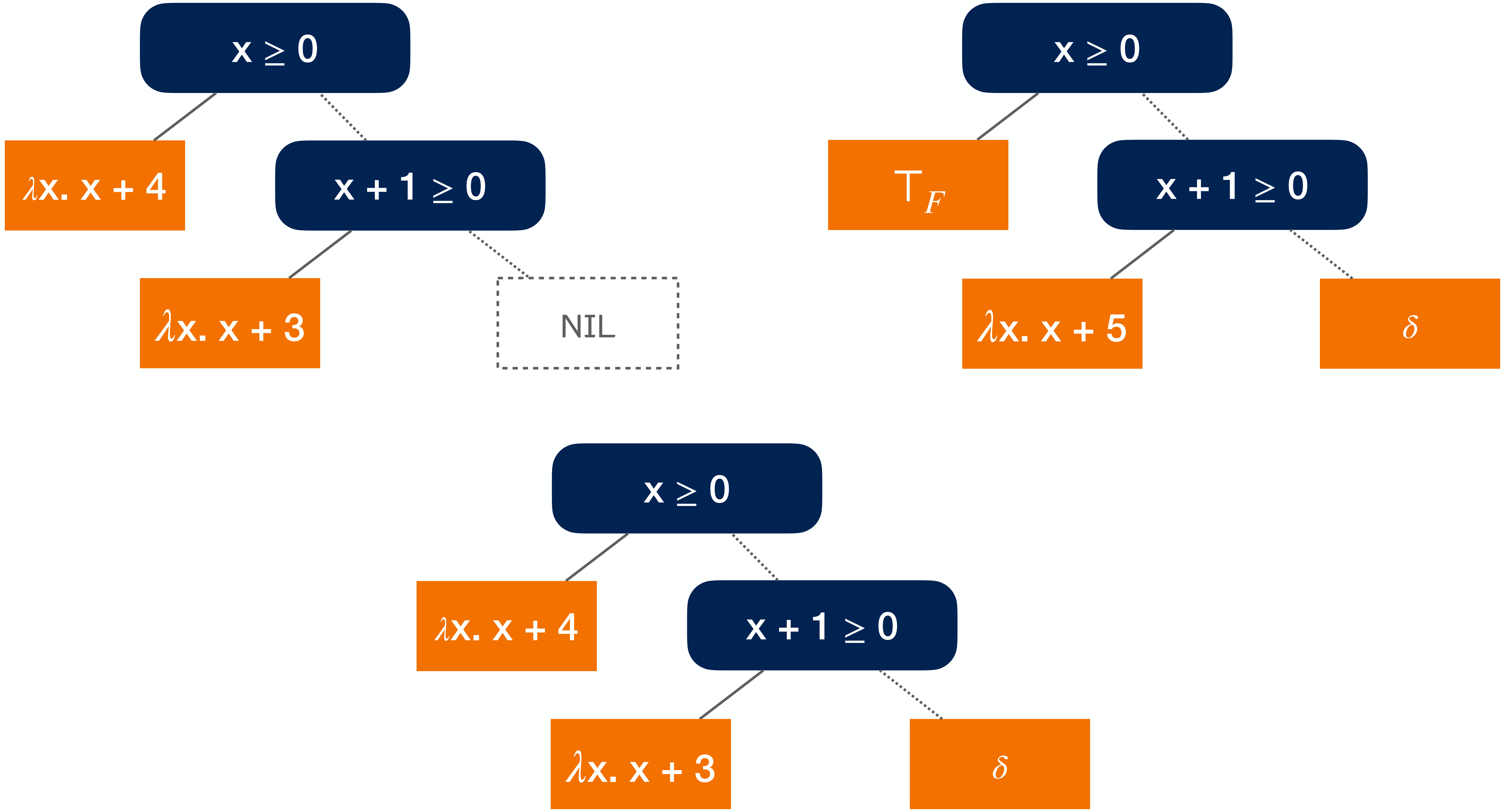
$f\colon \mathbb{Z}^{|\mathbb{V}|} \rightarrow \mathbb{N}$

$\bot_F$ $\qquad$ $\top_F$

Piecewise-Defined Ranking Functions Abstract Domain

Resilience Join

Example



Piecewise-Defined Ranking Functions Abstract Domain

Assignments

$\overleftarrow{\text{ASSIGN}}_A[\![X \leftarrow \textbf{input}]\!]$

- Base case ($\boxed{f}$)

Apply $\overleftarrow{\text{ASSIGN}}_F[\![X \leftarrow \textbf{input}]\!][\alpha_C(C)]$ on the <u>defined</u> leaf nodes

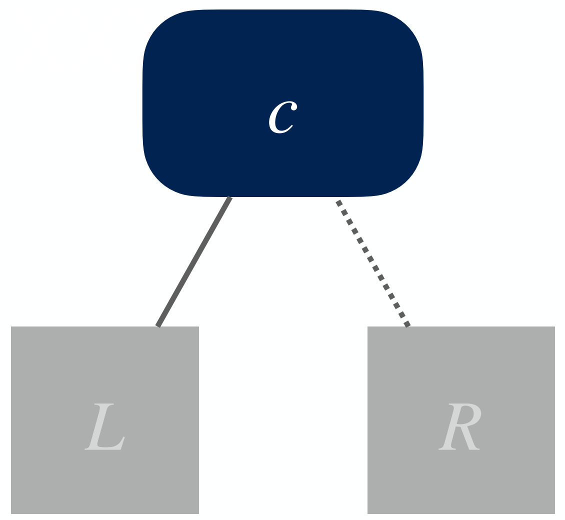
$$\overleftarrow{\text{ASSIGN}}_F[\![X \leftarrow \textbf{input}]\!][D](f) \stackrel{\text{def}}{=} \begin{cases} \bar{f} & \bar{f} \in \mathscr{F} \setminus \{ \perp_F , \top_F \} \\ \top_F & \text{otherwise} \end{cases} \qquad f \in \mathscr{F} \setminus \{ \perp_F , \top_F \}$$

where $\bar{f}(\ldots, X_i, X, \ldots) \stackrel{\text{def}}{=} \max\{f(\ldots, \rho(X_i), v, \ldots) + 1 \mid \rho \in \gamma_D(R) \wedge v \in E[\![e]\!]\rho\}$
and $R \stackrel{\text{def}}{=} \overleftarrow{\text{ASSIGN}}_D[\![X \leftarrow [-\infty, +\infty]]\!]D$

Piecewise-Defined Ranking Functions Abstract Domain

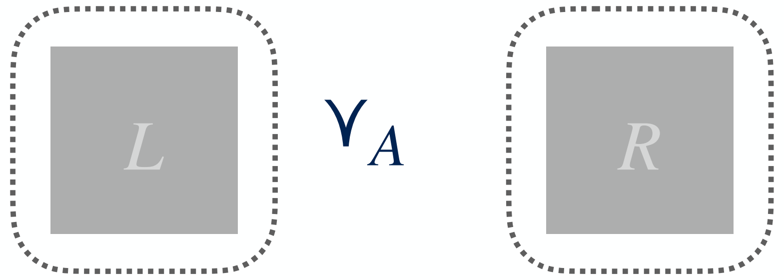
Assignments

$\overleftarrow{\mathrm{ASSIGN}}_A[\![X \leftarrow \textbf{input}]\!]$

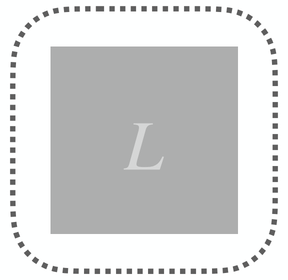
- 

Convert $\overleftarrow{\mathrm{ASSIGN}}_D[\![X \leftarrow [-\infty, +\infty]]\!](\alpha_C(\{c\})$ **and** $\overleftarrow{\mathrm{ASSIGN}}_D[\![X \leftarrow [-\infty, +\infty]]\!](\alpha_C(\{\neg c\})$ **into sets** I **and** J **of linear constraints** *in canonical form*

case ① $I = J = \varnothing$



case ② $I = \varnothing \wedge \bot_C \in J$



case ③ $\bot_C \in I \wedge J = \varnothing$

case ④

1. perform **tree pruning** on  and 
2. join the results with the **approximation join** $\curlyvee_A$

Piecewise-Defined Ranking Functions Abstract Domain

Assignments

$\overleftarrow{\text{ASSIGN}}_A[\![X \leftarrow e]\!]$

- Base case (f)

Apply $\overleftarrow{\text{ASSIGN}}_F[\![X \leftarrow e]\!][\alpha_C(C)]$ on the defined leaf nodes

the value of e (and thus of X) depends from the input values read by the program

$$\overleftarrow{\text{ASSIGN}}_F[\![X \leftarrow e]\!][D](f) \stackrel{\text{def}}{=} \begin{cases} \bar{f} & \text{TAINT}[\![e]\!] \wedge \bar{f} \in \mathcal{F} \setminus \{\bot_F, \top_F\} \\ \underline{f} & \neg\text{TAINT}[\![e]\!] \wedge \underline{f} \in \mathcal{F} \setminus \{\bot_F, \top_F\} \\ \top_F & \text{otherwise} \end{cases} \qquad f \in \mathcal{F} \setminus \{\bot_F, \top_F\}$$

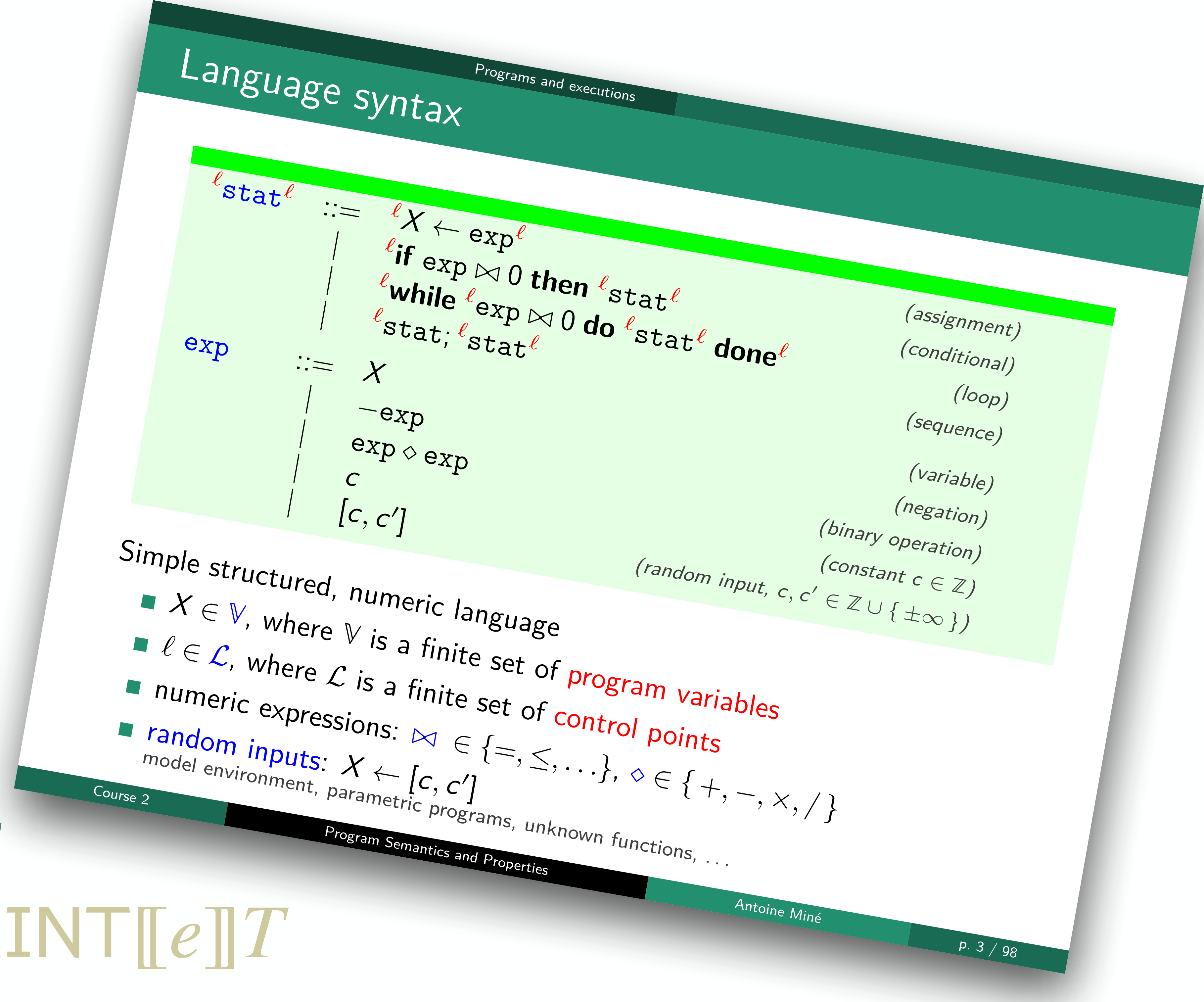
where $\bar{f}(\ldots, X_i, X, \ldots) \stackrel{\text{def}}{=} \max\{f(\ldots, \rho(X_i), v, \ldots) + 1 \mid \rho \in \gamma_D(R) \wedge v \in E[\![e]\!]\rho\}$

$\underline{f}(\ldots, X_i, X, \ldots) \stackrel{\text{def}}{=} \min\{f(\ldots, \rho(X_i), v, \ldots) + 1 \mid \rho \in \gamma_D(R) \wedge v \in E[\![e]\!]\rho\}$

and $R \stackrel{\text{def}}{=} \overleftarrow{\text{ASSIGN}}_D[\![X \leftarrow e]\!]D$

Simple Taint Analysis

$\mathsf{TAINT}[\![\mathrm{stat}]\!] : \mathscr{P}(\mathbb{X}) \rightarrow \mathscr{P}(\mathbb{X})$



$$\mathsf{TAINT}[\![{}^{\ell}X \leftarrow \mathbf{input}]\!]T \stackrel{\text{def}}{=} T \cup \{X\}$$

$$\mathsf{TAINT}[\![{}^{\ell}X \leftarrow e]\!]T \stackrel{\text{def}}{=} \begin{cases} T \cup \{X\} & \mathsf{TAINT}[\![e]\!]T \\ T \setminus \{X\} & \text{otherwise} \end{cases}$$

assigned variables in s

$$\mathsf{TAINT}[\![\mathbf{if}\ {}^{\ell}\ e \bowtie 0\ \mathbf{then}\ s]\!]T \stackrel{\text{def}}{=} \begin{cases} \mathsf{TAINT}[\![s]\!]T \cup \mathsf{LHS}(s) \cup T & \mathsf{TAINT}[\![e]\!]T \\ \mathsf{TAINT}[\![s]\!]T \cup T & \text{otherwise} \end{cases}$$

$$\mathsf{TAINT}[\![\mathbf{while}\ {}^{\ell}\ e \bowtie 0\ \mathbf{do}\ s\ \mathbf{done}]\!]T \stackrel{\text{def}}{=} \mathsf{lfp}^{\subseteq}_{T} \left(\lambda X.\begin{cases} \mathsf{TAINT}[\![s]\!]X \cup \mathsf{LHS}(s) \cup X & \mathsf{TAINT}[\![e]\!]X \\ \mathsf{TAINT}[\![s]\!]X \cup X & \text{otherwise} \end{cases} \right)$$

$$\mathsf{TAINT}[\![s_1; s_2]\!]T \stackrel{\text{def}}{=} \mathsf{TAINT}[\![s_2]\!](\mathsf{TAINT}[\![s_1]\!]T)$$

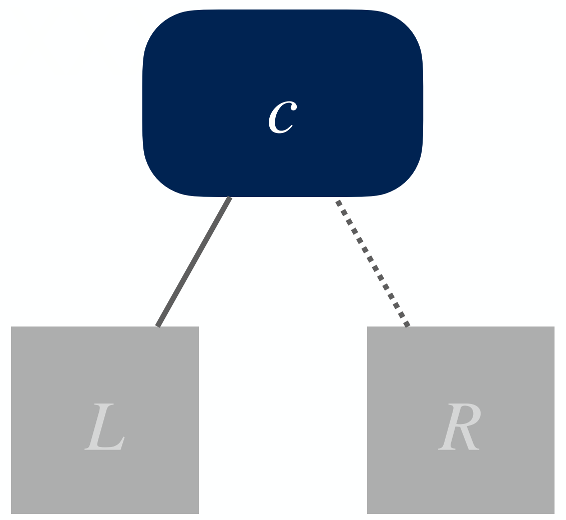
variables in e

$$\mathsf{TAINT}[\![e]\!]T \stackrel{\text{iff}}{\Leftrightarrow} \mathsf{VARS}(e) \cap T \neq \emptyset$$

Piecewise-Defined Ranking Functions Abstract Domain

Assignments

$\overleftarrow{\text{ASSIGN}}_A[\![X \leftarrow e]\!]$

- 

Convert $\overleftarrow{\text{ASSIGN}}_D[\![X \leftarrow e]\!](\alpha_C(\{c\})$ **and** $\overleftarrow{\text{ASSIGN}}_D[\![X \leftarrow e]\!](\alpha_C(\{\neg c\})$ **into sets** I **and** J **of linear constraints** *in canonical form*

case ① $I = J = \emptyset$

$L \curlyvee_A R$

case ② $I = \emptyset \wedge \bot_C \in J$

L

case ③ $\bot_C \in I \wedge J = \emptyset$

R

case ④

1. perform **tree pruning** on L and R
2. join the results with $\curlyvee_A$ (if $\text{TAINT}[\![e]\!]$) or $\underline{\vee}_A$ (if $\neg\text{TAINT}[\![e]\!]$)

Abstract Termination Resilience Semantics

Piecewise-Defined Ranking Functions Abstract Domain

For each program instruction stat, we define a transformer $\mathcal{R}^{\#}_{TR}[\![\text{stat}]\!]: \mathcal{A} \to \mathcal{A}$:

- $\mathcal{R}^{\#}_{TR}[\![{}^{\ell}X \leftarrow \textbf{input}]\!]t \overset{\text{def}}{=} \overleftarrow{\text{ASSIGN}}_A[\![X \leftarrow \textbf{input}]\!]t$
- $\mathcal{R}^{\#}_{TR}[\![{}^{\ell}X \leftarrow e]\!]t \overset{\text{def}}{=} \overleftarrow{\text{ASSIGN}}_A[\![X \leftarrow e]\!]t$
- $\mathcal{R}^{\#}_{TR}[\![\textbf{if}\ {}^{\ell}\ e \bowtie 0\ \textbf{then}\ s]\!]t \overset{\text{def}}{=} \begin{cases} \text{FILTER}_A[\![e \bowtie 0]\!](\mathcal{R}^{\#}_{TR}[\![s]\!]t) \curlyvee_T \text{FILTER}_A[\![e \not\bowtie 0]\!]t & \text{TAINT}_A[\![e]\!] \\ \text{FILTER}_A[\![e \bowtie 0]\!](\mathcal{R}^{\#}_{TR}[\![s]\!]t) \veebar_T \text{FILTER}_A[\![e \not\bowtie 0]\!]t & \text{otherwise} \end{cases}$
- $\mathcal{R}^{\#}_{TR}[\![\textbf{while}\ {}^{\ell}\ e \bowtie 0\ \textbf{do}\ s\ \textbf{done}]\!]t \overset{\text{def}}{=} \text{lfp}^{\#}\overline{F}^{\#}_{TR}$

  where $\overline{F}^{\#}_{TR}(x) \overset{\text{def}}{=} \begin{cases} \text{FILTER}_A[\![e \bowtie 0]\!](\mathcal{R}^{\#}_{TR}[\![s]\!]x) \curlyvee_T \text{FILTER}_A[\![e \not\bowtie 0]\!](t) & \text{TAINT}_A[\![e]\!] \\ \text{FILTER}_A[\![e \bowtie 0]\!](\mathcal{R}^{\#}_{TR}[\![s]\!]x) \veebar_T \text{FILTER}_A[\![e \not\bowtie 0]\!](t) & \text{otherwise} \end{cases}$
- $\mathcal{R}^{\#}_{TR}[\![s_1; s_2]\!]t \overset{\text{def}}{=} \mathcal{R}^{\#}_{TR}[\![s_1]\!](\mathcal{R}^{\#}_{TR}[\![s_2]\!]t)$

Abstract Termination Resilience Semantics

Piecewise-Defined Ranking Functions Abstract Domain

Definition

The **abstract termination resilience semantics** $\mathcal{R}^{\#}_{TR}[\![\mathsf{stat}^{\ell}]\!] \in \mathcal{A}$ of a program stat^{ℓ} is:

$$\mathcal{R}^{\#}_{TR}[\![\mathsf{stat}^{\ell}]\!] \stackrel{\text{def}}{=} \mathcal{R}^{\#}_{TR}[\![\mathsf{stat}]\!](\mathsf{LEAF}: \lambda X_1, \ldots, X_k.0)$$

where $\mathcal{R}^{\#}_{TR}[\![\mathsf{stat}]\!] : \mathcal{A} \to \mathcal{A}$ is the abstract termination resilience semantics of each instruction stat

Theorem (Soundness)

$$\mathcal{R}_{TR}[\![\mathsf{stat}^{\ell}]\!] \preccurlyeq \gamma_A(\mathcal{R}^{\#}_{TR}[\![\mathsf{stat}^{\ell}]\!])$$

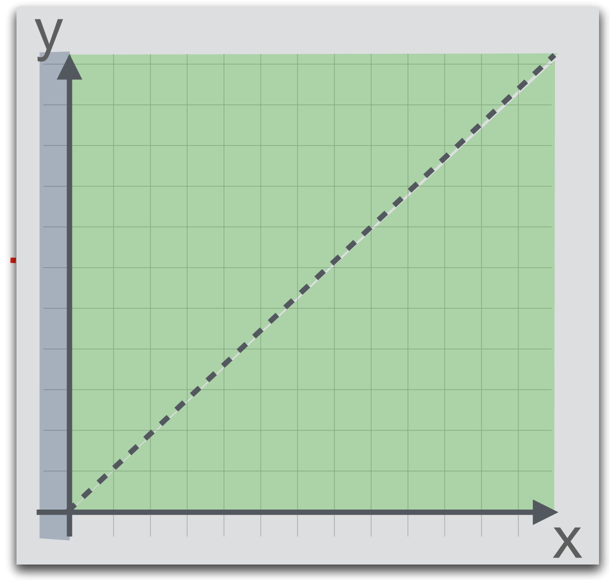
Corollary (Soundness)

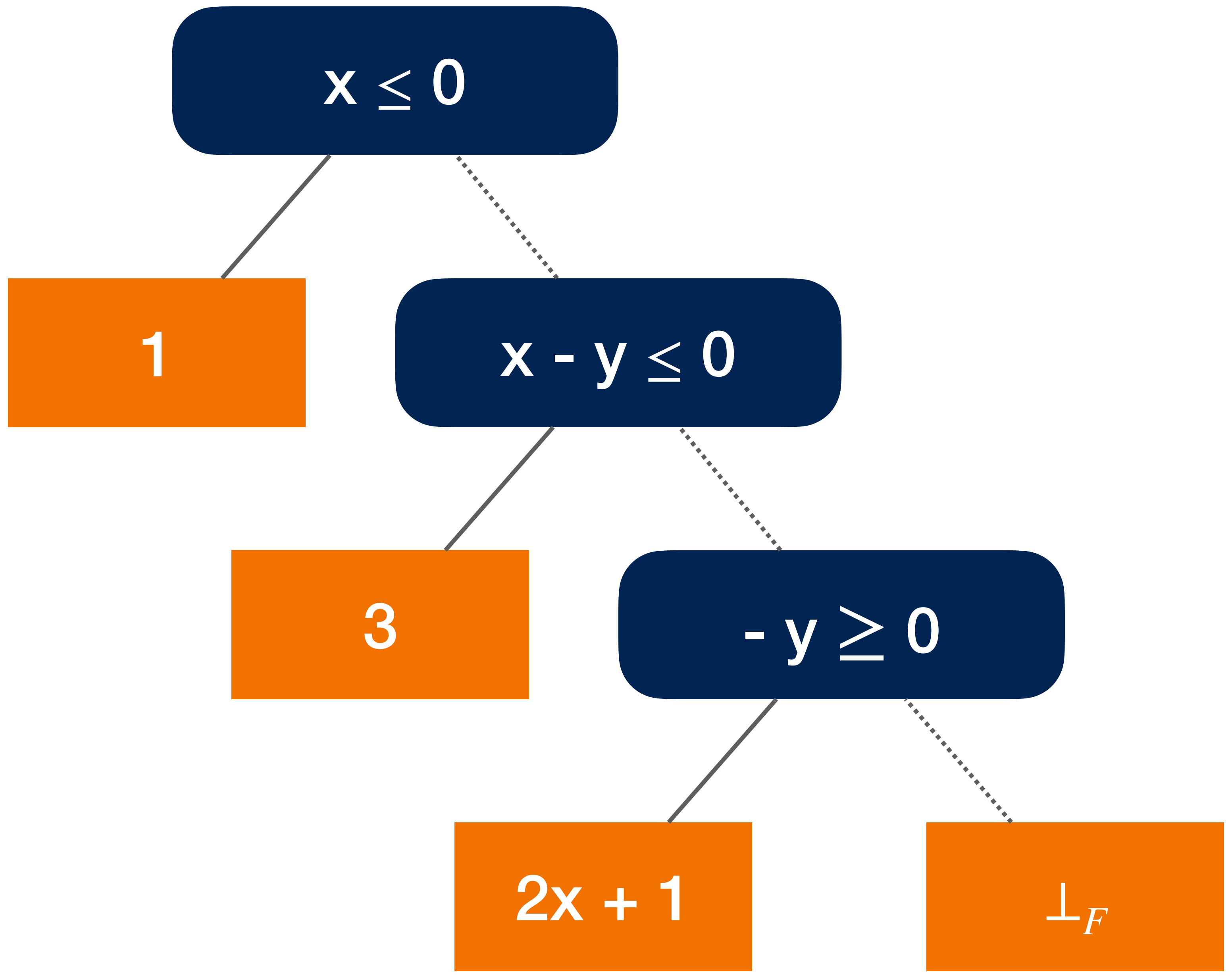
A program stat^{ℓ} **satisfies termination resilience** for traces starting from a set of initial states $\mathcal{I}$, i.e., $\mathcal{M}_{\infty}[\![\mathsf{stat}^{\ell}]\!](\mathcal{I}) \in \mathsf{TerminationResilience}$, if $\mathcal{I} \subseteq \mathsf{dom}(\gamma_A(\mathcal{R}^{\#}_{TR}[\![\mathsf{stat}^{\ell}]\!]))$

Abstract Termination Resilience Semantics

Example

[1]x ← [-∞, +∞]
[2]y ← **input**
while [3](x > 0) **do**
    [4]x ← x - y
od[5]

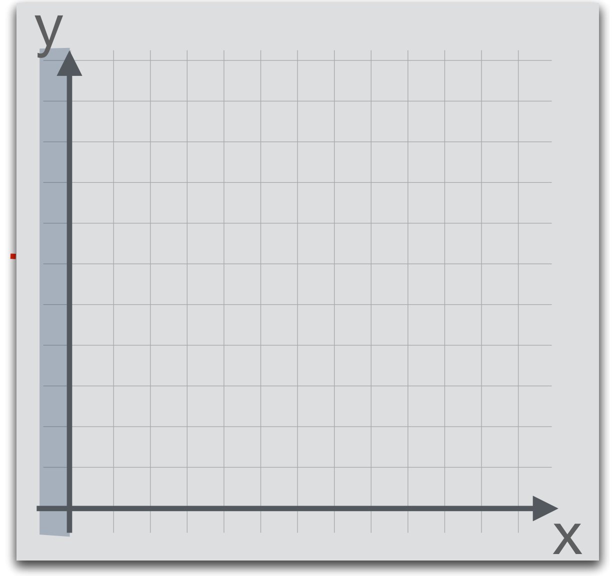


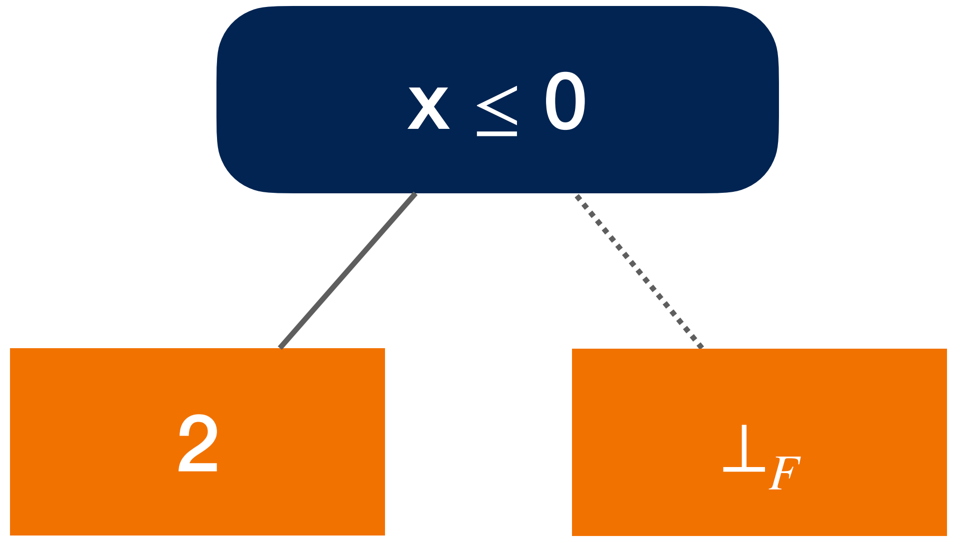


Abstract Termination Resilience Semantics

Example

[1]x ← [-∞, +∞]
[2]y ← **input**
while [3](x > 0) **do**
  [4]x ← x - y
od[5]





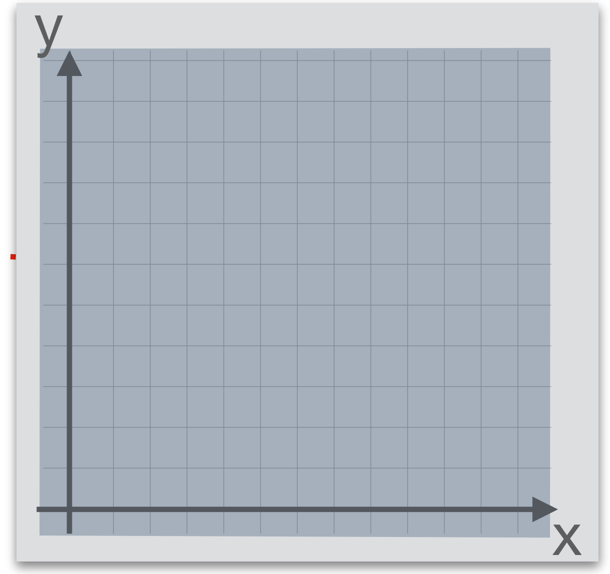
$\overleftarrow{\text{ASSIGN}}_A[\![y \leftarrow \textbf{input}]\!]$

Abstract Termination Resilience Semantics

Example

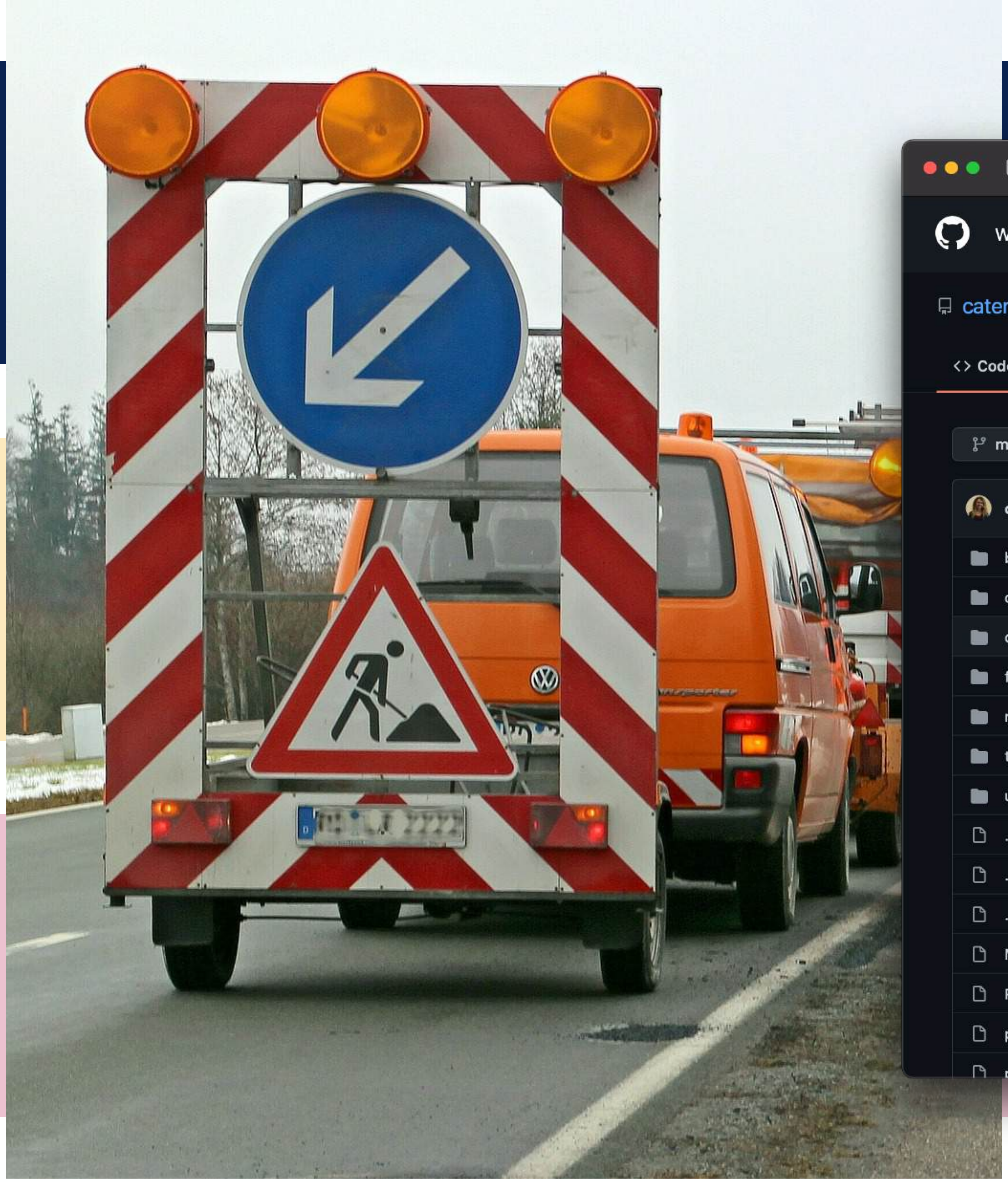
$\overleftarrow{\text{ASSIGN}}_A[\![x \leftarrow [-\infty, +\infty]]\!]$

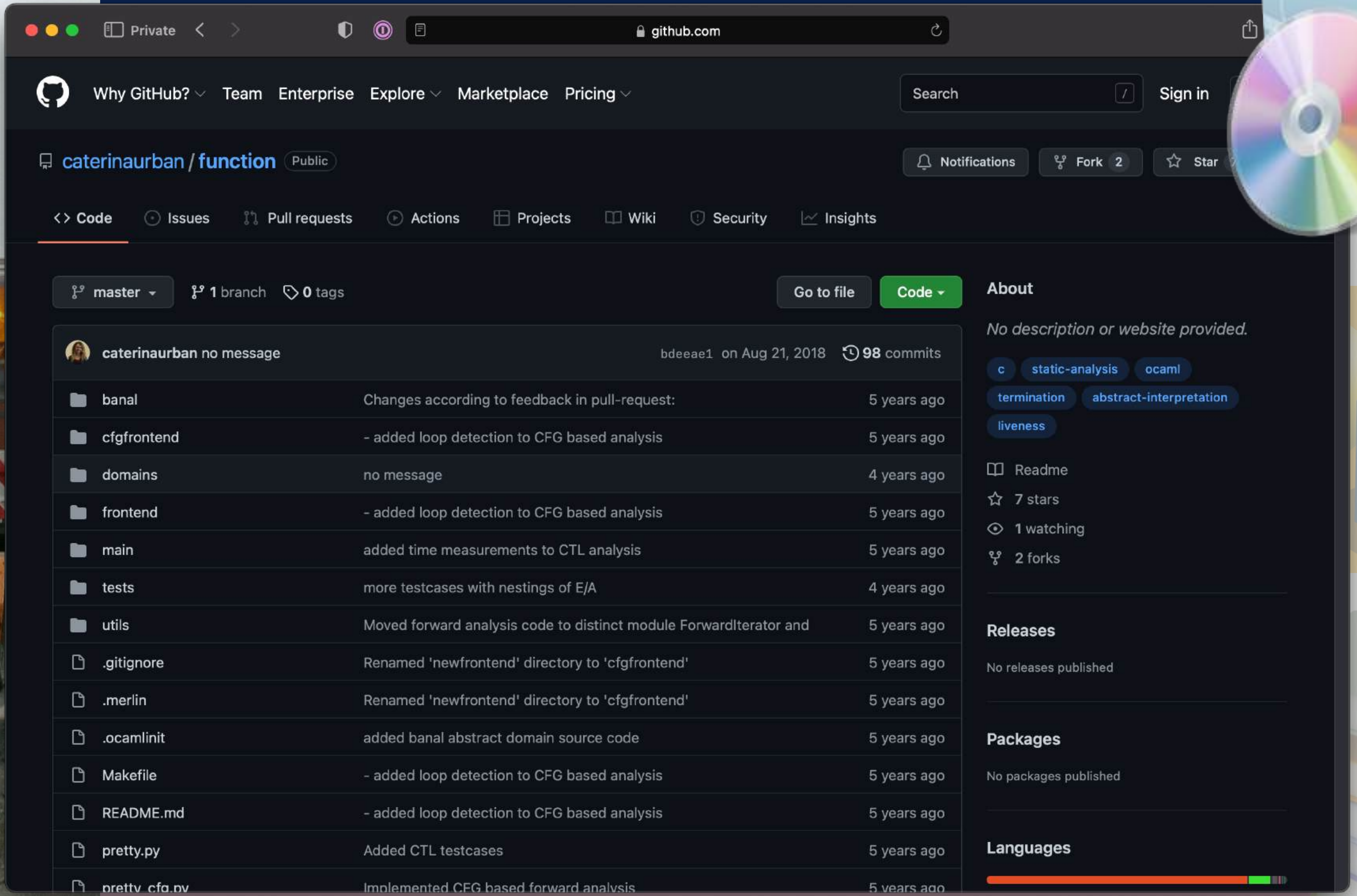
1x ← [-∞, +∞]
2y ← **input**
while 3(x > 0) **do**
    4x ← x - y
od5



3

Implementation





Bibliography

[Moussaoui24] **Naïm Moussaoui Remil and Caterina Urban**. Termination Resilience Static Analysis. Draft, 2024.

if interested, ask me for a **draft**